

# GoDP: Globally Optimized Dual Pathway system for facial landmark localization in-the-wild

Yuhang Wu, Shishir K. Shah, Ioannis A. Kakadiaris

{ywu35,sshah,ikakadia}@central.uh.edu

Computational Biomedicine Lab  
Department of Computer Science, University of Houston  
4849 Calhoun Road, Houston, TX, 77004

---

## Abstract

Facial landmark localization is a fundamental module for face recognition. Current common approach for facial landmark detection is cascaded regression, which is composed by two steps: feature extraction and facial shape regression. Recent methods employ deep convolutional networks to extract robust features in each step and the whole system could be regarded as a deep cascaded regression architecture. Unfortunately, this architecture is problematic. First, parameters in the networks are optimized from a greedy stage-wise perspective. Second, the network cannot efficiently merge landmark coordinate vectors with 2D convolutional layers. Third, the facial shape regression relies on a feature vector generated from the bottom layer of the convolutional neural network, which has recently been criticized for lacking spatial resolution to accomplish pixel-wise localization tasks. We propose a globally optimized dual-pathway system (GoDP) to handle the optimization and precision weaknesses of deep cascaded regression without resorting to high-level inference models or complex stacked architecture. This end-to-end system relies on distance-aware softmax functions and dual-pathway proposal-refinement architecture. The proposed system outperforms the state-of-the-art cascaded regression-based methods on multiple in-the-wild face alignment databases. Experiments on face identification demonstrate that GoDP significantly improves the quality of face frontalization in face recognition.

**Keywords:** Deep Learning, facial landmark localization, face alignment, face recognition

---

## 1. Introduction

With the exponential increase in personal images in PC and mobile devices, highly accurate, efficient, and fully automatic facial landmark annotators are highly needed. Facial landmark annotation is a problem of localizing salient facial landmarks (e.g., eye corners, nose tip) on human face. If we assume that there are  $L$  facial landmarks whose locations need to be estimated, then the target space of this problem is a  $2L \times 1$  feature vector where each attribute corresponds to the horizontal or vertical coordinate of a specific landmark in a given image. In biometric research, facial landmark annotation is a fundamental module in face recognition, face tracking, and 3D facial model recovery. These systems rely on landmarks to construct semantic correspondences between facial images. In real-world deployment, facial landmark detection may become a bottleneck that constrains these systems to achieve optimum performance.

Even though the boundaries of landmark detection have been consistently pushed forward in past years, localizing landmarks under unconstrained conditions (detecting landmarks under large head pose deviations, facial expression variations, illumination changes, and face occlusions) still remains challenging due to multiple difficulties.

Current common approaches for facial key-point localization are based on cascaded regression [1, 2, 3, 4]. While cascaded regression is a useful framework for face alignment, sev-

eral challenges need to be addressed when deriving a deep architecture. First, current deep cascaded regression is greedily optimized per each stage. The learned mapping  $\mathbb{R}^t$  is not end-to-end optimal with respect to the global shape increment. When training a new mapping  $\mathbb{R}^t$  for stage  $t$ , fixing the network parameters of previous stages leads to a stage-wise sub-optimal solution. Different from cascaded face detection, which can be easily formulated into a globally optimal structure [5], gradients that back-propagate from the later stages of the network are blocked due to reinitialization of shape parameters between stages. The second challenge arises from shape-indexed features. Shape-indexed features are extracted based on landmark locations [1, 2]. However, how to effectively merge the information encoded in 1D coordinate vectors into a 2D image in an optimal way still remains an open problem. Even though some heuristic solutions (e.g., concatenating a 3D geometric map with RGB channels [3] or highlighting pixel blocks [6]) alleviate the problem somewhat, the solution is not optimal from a gradient back-propagation perspective since the pixel values in the newly generated maps are assigned based on external parameters. The third challenge comes from information flow in deep architecture. The deep representation generated at the bottom layer, while highly discriminative and robust for object/face representation, loses too much spatial resolution after many pooling and convolutional layers. As a result, it can-

not tackle pixel-level localization/classification tasks very well. This phenomenon was recently named in the image segmentation field as spatial-semantic uncertainty [7]. Because in most deep regressions [8, 9, 3, 4, 10], where  $\mathbb{R}^l$  solely relies on the deep representations generated by the latest layers of the networks, the precision of predictions may suffer from this structural limitation of deep networks.

To tackle the aforementioned challenges in deep cascaded regression models, we propose a globally optimized dual pathway (GoDP) system where all inferences are conducted on 2D score maps to facilitate gradient back-propagation. Because there are very few landmark locations activated on the 2D score maps, we propose a distance-aware softmax function (DSL) that reduces the false alarms in the 2D score maps. To solve the spatial-semantic uncertainty problem of deep architecture, we propose a dual pathway model where shallow and deep layers of the network are jointly forced to maximize the possibility of a highly specific candidate region. As a result, our facial key-points localization model achieved state-of-the-art performance on multiple challenging databases. To further demonstrate the contribution of our work, we embedded the proposed landmark detector into a 3D-aided face recognition system. Our results shows that our method significantly improves the quality of 3D face frontalization in challenging poses. In addition, we demonstrate that with good face alignment, shallow features [11] could outperform deep representations (e.g., Openface [12], VGG face descriptor [13]).

The key contributions of our work include:

- A deep network that is able to generate high quality 2D score maps for key-points localization without stacked architecture.
- A new loss function designed for reducing false alarms in the 2D score maps.
- A heavily supervised proposal-refinement architecture to discriminatively extract spatial-semantic information from the deep network.
- Significantly improved quality of face frontalization in 3D-aided face recognition.

The rest of this paper is organized as follows. In Section 2, we present the related work in deep cascaded regression and pixel-labeling. In Section 3, we introduce the proposed deep architecture and three critical components of this architecture. In Section 4, we evaluate the proposed method in challenging databases of face alignment and identification. In Section 5, we present our conclusions.

## 2. Related Work

Most of the deep architectures used for face alignment are extended from the framework of cascaded regression. Sun *et al.* [8] first employed an AlexNet-like architecture to localize five fiducial points on faces. Later, Zhang *et al.* [9] proposed

a multi-task framework demonstrating that a more robust landmark detector can be built through joint learning with correlated auxiliary tasks, such as head pose and facial expression, which outperformed several shallow architectures [14, 15, 16, 17]. To conquer facial alignment problems under arbitrary head poses, Zhu *et al.* [2] and Jourabloo *et al.* [4] employed a deformable 3D model to jointly estimate facial poses and shape coefficients on-line. These deep cascaded regression methods outperform multiple state-of-the-art shallow structures [18, 19, 20], and achieved remarkable performance on less controlled databases such as AFLW [21] and AFW [22]. To improve facial landmark localization accuracy, some researchers [8, 23, 24, 25] have employed local deep networks to localize facial landmarks based on facial patches. These networks rely strongly on the accuracy of a global network to select correct landmark candidate regions before being deployed. Once the global network fails, the local networks cannot fully correct the accumulated errors. A common point of the previous architectures is that they require model re-initialization when switching stages/networks. As a result, the parameters of each stage/network are optimized from a greedy stage-wise perspective, which is inefficient and suboptimal.

Inspired by recent work in human pose estimation and face alignment [26, 27, 6], we employ 2D score maps as the targets for inference. This modification enables gradients back-propagation between stages, allows 2D feedback loops, and hence delivers an end-to-end model. In this new family of methods for key-points localization, recent works [6, 28] rely on a DeconvNet [29] architecture to localize facial landmarks. Even though they obtain impressive results by integrating the estimation with recurrent neural networks, the quality of face alignment is intrinsically limited by the precision weakness of the DeconvNet. Wei *et al.* [27] proposed convolutional pose machine (CPM), which employs a stacked cascaded architecture to refine body key-point predictions. This cascaded structure has multiple subnetworks that gradually minimize the residual errors of score maps. It is an end-to-end deep model. However, the input of each subnetwork in [27] is the original image, which ends up with a heavy and redundant architecture that is difficult for small scale deployment. Bulat *et al.* [26] employs a two-stage convolutional aggregation architecture where a CNN detector is trained first, then a CNN regressor is learned based on the input of both shallow-level and deep-level features of the CNN detector to further boost the accuracy of landmark localization. The two-stage architecture contains multiple convolutional layers with large filter size (the whole network cost nearly 12GB GPU memory for processing one image in testing), which is too heavy to be deployed in a small-scale GPU. In contrast, our dual-pathway architecture contains a single network which takes far less GPU memory and obtain better performance.

One fundamental challenge when employing 2D score maps for key-point localization is spatial-semantic uncertainty, which is critical but has not been the focus of previous works on face alignment. Ghiasi *et al.* [7] pointed out that features generated from the bottom layers of deep networks, although encoding semantic information which is robust to image and human iden-

tity variations, lack enough spatial resolution for tasks requiring pixel-level precision (*e.g.*, image segmentation, key-points localization). To tackle this problem, the authors of [6, 10] concatenated shallow-level convolutional layers to the latest convolutional layers before landmark regression. Newell *et al.* [30] proposed a heavily stacked structure by intensively aggregating shallow and deep convolutional layers to obtain better score map predictions. In image segmentation, Pinherio *et al.* [31] proposed a top-down refinement architecture which first generates robust but low-resolution score maps in a feedforward pass, then gradually refines the score maps in a top-down pass using features at lower-level layers. Although the aforementioned methods improve the deep networks' resolution for accomplishing pixel-level labeling tasks, concatenating and adding noised shallow-level features with deep-level features without regularization could be detrimental to the system's discriminative capability. Ghiasi *et al.* [7] proposed a Laplacian-pyramid-like architecture that refines the 2D score maps generated by the bottom layers by adding back features generated from top layers with the supervision of three soft-max loss layers, which provides more constraints to the refinement. In our work, we go a step further by introducing a well-controlled proposal and refinement architecture for key-point localization. By imposing an appropriate supervising signal, shallow-level features in our architecture are used to propose key-point candidates and deep-level features are responsible for refining the proposals and suppressing the false-alarms in the background. A new loss function is proposed to impose appropriate regularization over different network layers and guarantee the whole architecture works as expected. Experimental results show that our architecture outperforms adding-back structure employed by Newell *et al.* [30] in the DeconvNet architecture.

We observed that recently, more works in landmark and body joint localization rely on high-level inference models such as recurrent network [6, 28], conditional random fields [32], or other graphical models [33]. In this work, we do not resort to any high-level inference architecture and concentrate on improving a traditional deep network by better employing the information encoded in network layers. We provide a general and improved method to obtain a clear and discriminative score map. In probabilistic graphic models, a better unary term can be constructed with the proposed architecture, which can be jointly learned with any pair-wise terms to further boost the performance of the inference system.

### 3. Method

In this section, we first briefly review deep cascaded regression, then introduce three components of the proposed architecture. They are the basic elements that help us address multiple challenges in 2D score map-based inference. Then, we introduce our GoDP system.

#### 3.1. Deep cascaded regression

The intent of cascaded regression is to progressively minimize a difference  $\Delta S$  between a predicted shape  $\hat{S}$  and a ground-truth shape  $S$  in an incremental manner. This approach contains

$T$  stages, starting with an initial shape  $\hat{S}^0$ ; the estimated shape  $\hat{S}^t$  is gradually refined as:

$$\arg \min_{\mathbb{R}^t, \mathbb{F}^t} \sum_i \|\Delta S_i^t - \mathbb{R}^t(\mathbb{F}^t(\hat{S}_i^{t-1}, \mathbf{I}_i))\|_2^2; \quad (1)$$

$$\hat{S}_i^t = \hat{S}_i^{t-1} + \Delta \hat{S}_i^{t-1} \quad (2)$$

where  $i$  iterates over all training images. The estimated facial shape for image  $\mathbf{I}_i$  in stage  $t$  is denoted by  $\hat{S}_i^t$ ; usually  $\hat{S}_i^t$  can be represented as a  $2L \times 1$  vector. The number of facial key-points is denoted by  $L$ . The function  $\mathbb{F}^t(\hat{S}_i^{t-1}, \mathbf{I}_i)$  is a mapping from image space to feature space. Because the obtained features are partially determined by  $\hat{S}_i^{t-1}$ , these features are called 'shape-indexed features'. The function  $\mathbb{R}^t(\cdot)$  is a learned mapping from feature space to target parameter space. In deep cascaded regression [8, 9, 3, 4], function  $\mathbb{F}^t(\cdot)$  can be used to denote all operations before the last fully connected layer. This mapping  $\mathbb{R}^t(\cdot)$  represents the operations in the last fully connected layer whose input is an arbitrary dimensions feature vector  $\phi_i^t$  and output is the target parameter space.

The main problem of this architecture is the current deep cascaded regression is greedily optimized per each stage. The learned mapping  $\mathbb{R}^t$  is not end-to-end optimal with respect to the global shape increment. When training a new mapping  $\mathbb{R}^t$  for stage  $t$ , fixing the network parameters of previous stages leads to a stage-wise suboptimal solution.

#### 3.2. Optimized progressive refinement

Due to optimization problems in traditional deep cascaded architecture, a global optimization model is highly needed. However, the main difficulty in converting a cascaded regression approach into a globally optimized architecture is to back-propagate gradients between stages, where shape was usually used as a prior to initialize new cascaded stages. In our work, we bypass the problem by representing landmark locations  $\hat{S}_i^t$  through 2D score maps  $\Psi^t$  (we omit the index  $i$  for clarity), where information of landmark positions is summarized into probability values that indicate the likelihood of the existence of landmarks. In our work, the tensor  $\Psi^t$  denotes  $(KL+1) \times W \times H$  score maps in stage  $t$ , where  $L$  is the number of landmarks,  $K$  is the number of subspaces (which will be introduced later),  $W$  and  $H$  are the width and height of the score maps. The extra  $(KL+1)^{th}$  channel indicates the likelihood that a pixel belongs to background. Through this representation, gradients can pass through the score maps and be back-propagated from the latest stages of cascaded model. Another insight of employing 2D probabilistic score maps is that these outputs can be aggregated and summarized with convolutional features and create feedback paths [34], which can be represented as follows:

$$\Psi^0 = \mathbb{F}_o^0(\mathbf{I}) \quad (3)$$

$$\mathbb{F}_b^{t-1}(\mathbf{I}, \Psi^{t-1}) = \mathbb{F}_a^{t-1}(\mathbf{I}) \uplus \Psi^{t-1} \quad (4)$$

$$\Delta \Psi^{t-1} = \mathbb{F}_c^{t-1}(\mathbb{F}_b^{t-1}(\mathbf{I}, \Psi^{t-1})) \quad (5)$$

$$\Psi^t = \Psi^{t-1} + \Delta \Psi^{t-1} \quad (6)$$

where  $\oplus$  denotes a layerwise concatenation in convolutional neural networks,  $\mathbb{F}_o^0(\mathbf{I})$  represents the first  $\Psi^0$  generated from  $\mathbf{I}$  after passing through several layers in convolutional neural network,  $\mathbb{F}_a^{t-1}(\cdot)$ ,  $\mathbb{F}_b^{t-1}(\cdot)$  and  $\mathbb{F}_c^{t-1}(\cdot)$  indicate different network operations with different parameter settings.

Equations 4 to 6 can be regarded as an iterative error feedback path [34]. Four of these paths are included in our structure as shown in Fig. 2. Through the feedback paths, score maps generated by each stage can be directly concatenated with other convolutional layers through Eq. (4), which behaves as ‘shape-indexed feature’. In contrast to [3] and [6], where score maps employed in the feedback paths are determined and synthesized through external parameters, in our architecture,  $\Psi^{t-1}$  in Eq. (4) is fully determined by the parameters inside the network based on Eq. (5) and Eq. (6). Therefore, our face alignment model can be optimized globally.

### 3.3. 3D pose-aware score map

To model complex appearance-shape dependencies on human face across pose variations, unlike recent works that employ a deformable shape model [4, 2], our model implicitly encodes 3D constraints. We found that pose is a relative concept whose actual value is susceptible to the sampling region, facial expression, and other factors. As a result, it is very hard to learn an accurate and reliable mapping from image to the pose parameters without considering fiducial points’ correspondence. Instead of estimating pose parameters [35, 20, 4, 2] explicitly, we regard pose as a general domain index that encodes multi-modality variations of facial appearance. Specifically, we used  $K$  score maps to indicate each landmark location, where  $K$  corresponds to the number of partitions of head pose. For each image, one out of  $K$  score maps is activated for each landmark. In this way, the network automatically encodes contextual information between appearance of landmarks under different poses. At the final stage, the  $K$  score maps are merged into one score map through element-wise summation. In our implementation,  $K$  is equal to 3, and the subspace partition is determined by yaw variations.

### 3.4. Distance-aware softmax loss

Soft-max loss function has been widely used in solving pixel-labeling problems in human joint localization [27, 36], image segmentation [29, 7], and recently, facial key-point annotation [6]. One limitation of using softmax for key-point localization is that the function treats pixel-labeling as an independent classification problem, which does not take into account the distance between the labeling pixel and the ground-truth key-points. As a result, the loss function will assign equal penalty to the regions that lie very close to a key-point and also the regions on the border of an image, which should not be classified as landmark candidates. Another drawback of this loss function is that it assigns equal weights to negative and positive samples, which may lead the network to converge into a local minimum, where every pixel is marked as background. This is a feasible solution from an energy perspective because the active pixels in the score maps are so sparse (only 1 pixel is marked as key-point per score map in maximum) that their weights play a very

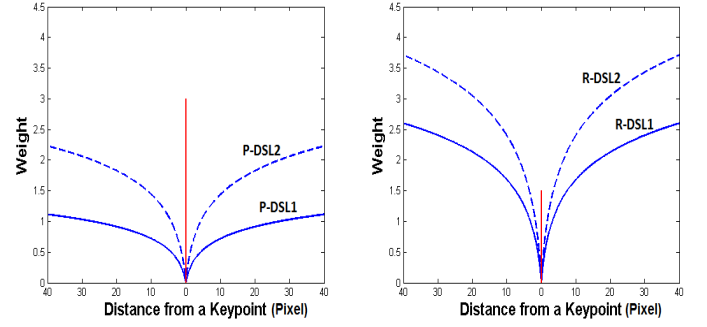


Figure 1: The shape of the distance-aware softmax loss (DSL) employed in the decision pathway. The transformations of the functions after increasing the values of  $\beta$  are visualized through the dashed lines. The straight red lines indicate the cost of misclassifying a key-point pixel to a background pixel, while the blue lines indicate the cost of misclassifying a background pixel to a key-point pixel. (L) DSL for proposal, (R) DSL for refinement.

small role in the loss function compared to the background pixels. To solve the above-mentioned two problems, we modified the original loss function as follows. First, we assign larger cost when the network classifies a keypoint pixel into background class; this helps the model stay away from local minima. Second, we assign different cost to the labeled pixels according to the distance between the labeled pixels and other key-points, which makes the model aware of distances. This loss function can be formulated as follows:

$$\sum_x \sum_y m(x, y) w \sum_k t_k(x, y) \log\left(\frac{e^{\psi_k(x, y)}}{\sum_{k'} e^{\psi_{k'}(x, y)}}\right) \quad (7)$$

$$w = \begin{cases} \alpha, & k \in \{1 : KL\} \\ \beta \cdot \log(d((x, y), (x', y')) + 1), & k = KL + 1 \end{cases} \quad (7a) \quad (7b)$$

where  $(x, y)$  are locations,  $k \in \{1 : KL + 1\}$  is the index of classes,  $\psi_k(x, y)$  is the pixel value at  $(x, y)$  in the  $k^{th}$  score map of  $\Psi$ ,  $t_k(x, y) = 1$  if  $(x, y)$  belongs to class  $k$ , and 0 otherwise. The binary mask  $m(x, y)$  is used to balance the amount of key-point and background pixels employed in training. The weight  $w$  controls the penalty of foreground and background pixel. For a foreground pixel, we assign a constant weight  $\alpha$  to  $w$ , whose penalty is substantially larger than nearby background pixels. While for a background pixel, the distance  $d((x, y), (x', y'))$  between the current pixel  $(x, y)$  and a key-point  $(x', y')$  whose probability ranked the highest among the  $KL$  classes is taken into account. The result is that the loss function assigns the weights based on the distance between the current pixel and the most misleading foreground pixel among the score maps, which punishes false-alarms adaptively. In Eq. (7b), we used a log function (base 10) to transform the distance into a weight, and employed a constant  $\beta$  to control the magnitude of the cost. The shape of  $w$  is depicted in Fig. 1. As a result, discrimination between the background and foreground pixels is encouraged according to the distance between a labeled pixel and a

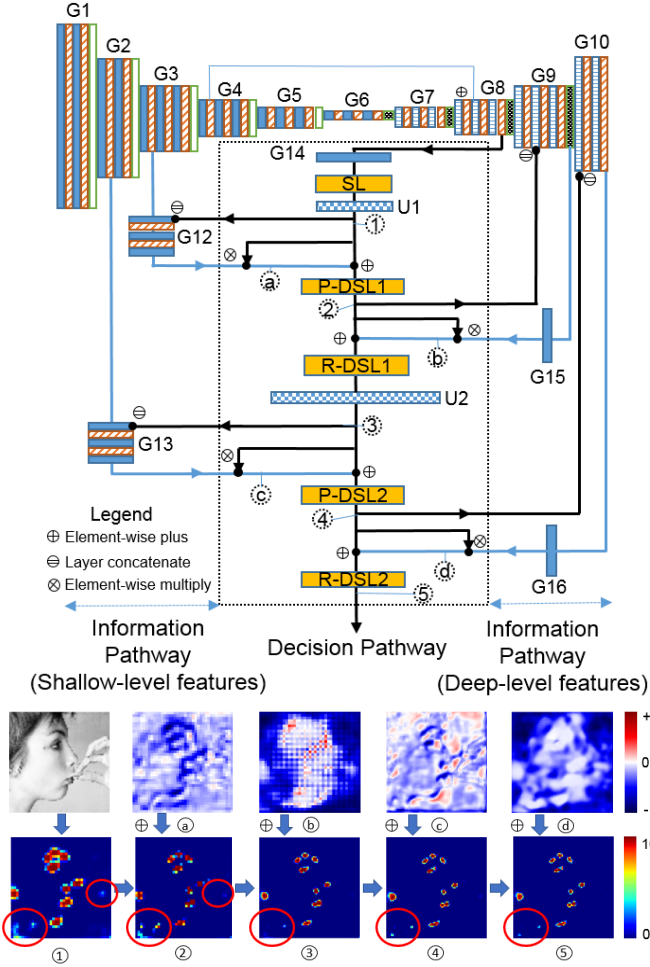


Figure 2: The architecture of the proposed globally optimized dual-pathway (GoDP) model. Based on a naive DeconvNet [29], we derive a precise key-point detector by discriminatively extracting spatial and semantic information from shallow and deep layers of the network. The framework is motivated by cascaded regression, which contains residual error corrections and error feedback loops. Moreover, GoDP is end-to-end trainable, fully convolutional, and optimized from a pixel labeling perspective instead of traditional regression. Under the architecture, we visualize the  $(KL + 1)^{th}$  score map sampled through the network, which indicates the probability of key-point locations. The letter and number below each score map indicate the corresponding position in the network architecture. We highlight background regions with red circles to indicate how the proposal and refinement technique deal with background noises (best viewed in color).

specific key-point. From a point of view of optimization, in back-propagation, since  $d((x, y), (x', y'))$  is independent from  $\psi_k(x, y)$ ,  $w$  will be a constant which can be directly computed through Eq. (7a) or Eq. (7b).

When training the network, we first replace Eq. (7b) with a constant term (represented as  $\beta$ , which is less than  $\alpha$ ) and train the network with this degraded DSL (represented as SL in Fig. 2) for the first six epochs. Then, Eq. (7b) is employed for further refinement. During training, inspired by curriculum learning [37], we gradually increase the value of  $\beta$  and encourage the network to discriminate pixels closer to the key-point locations.

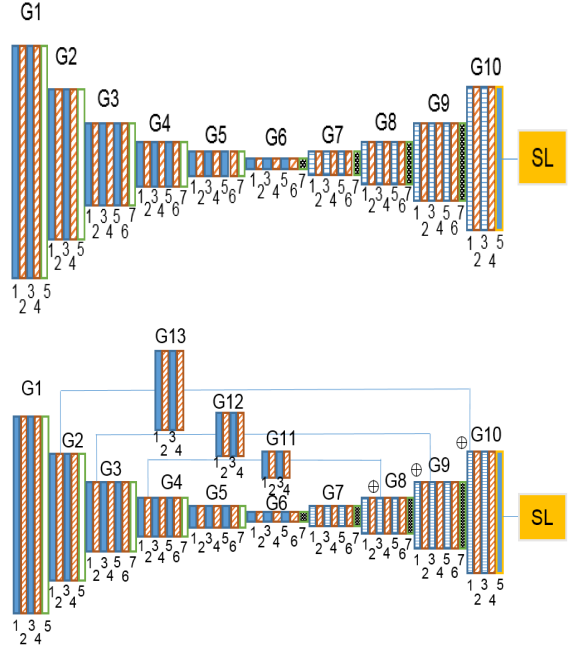


Figure 3: Baseline network settings: (T) DeconvNet, (B) DeconvNet with Hour-glass [30] connections.

### 3.5. Proposal and refinement in dual-pathway

To better exploit the spatial and semantic information encoded in a deep network, we propose a dual pathway architecture as shown in Fig. 2. Our network is designed based on DeconvNet [29]. It is a classical architecture widely used for pixel-wise labeling. The intuition of the network can be explained as a supervised auto-encoder, which employs an encoder to extract high-level representation from the original image, then transforms the representation to a target semantic domain through a decoder. Recently a variation has been employed for face alignment [6, 28]. To alleviate spatial information loss caused by max-pooling layers, in DeconvNet, unpooling is used for preserving the stimulus structure before deconvolution. Derived from DeconvNet [29], the unique design of our proposed architecture includes separate pathways used for generating discriminative features and making decisions. We designate them as “information pathway” and “decision pathway”. In the decision pathway, the depth of each layer is strictly kept as  $KL + 1$  where each channel corresponds to a score map  $\psi_k$ . In the information pathway, depths of layers are unconstrained to enrich task-relevant information.

**Features in the information pathway:** The design of the information pathway is built upon the findings that feature maps generated from the deep layers of the network contain robust information that is invariant to the changing of image conditions but lack enough resolution to encode exact key-point locations. While the feature maps of shallow layers contain enough spatial information to localize the exact position of each key-point, while they also contain a large amount of irrelevant noise. To handle this dilemma, we build a structure such that the features extracted from shallow layers are used to propose candidate re-

Table 1: Detailed experiment settings of our algorithm.

Evaluation Name	Training Set	# of Training Samples	Trained Model	Testing Set	# of Testing Samples	Point	Normalizing Factor	Settings
AFLW-PIFA	AFLW	3,901	<i>M1</i>	AFLW	1,299	21	Face Size	Following [2]
AFLW-Full	AFLW	20,000	<i>M2</i>	AFLW	4,386	19	Face Size	Following [2]
AFLW-F	AFLW	-	<i>M2</i>	AFLW	1,314	19	Face Size	Section 4.1
AFW	AFLW	-	<i>M2</i>	AFW	468	6 out of 19	Face Size	Section 4.1
UHDB31	AFLW	-	<i>M2</i>	UHDB31	1,617	9 out of 19	Face Size	Section 4.1

gions while the features extracted from deep layers help to filter out false alarms and provide structural constraints. This is accomplished by imposing different losses to supervise shallow-level and deep-level features generated from shallow and deep layers. We adjust the parameters of DSL in the decision pathway and enforce a large penalty when the shallow-level features fail to assign large positive probabilities to key-point locations, but give a smaller cost when they misidentify a background into a key-point candidate. This is a high detection rate policy to supervise shallow-level features. In contrast, we adopt a low false alarm policy to supervise deep-level features: we enforce high penalty when deep-level features misidentify a background pixel as key-point but slightly tolerate the error in the other way around. The results are shown in Fig. 2. After each shallow-level proposal, the contrast between background and foreground is increased, while, after each deep-level refinement the background noise is suppressed. As a result, the key-point regions are gradually shrunk and highlighted.

**Score maps in the decision pathway:** In the decision pathway, the tensor  $\Psi^0$  is first initialized with the output of  $2^{nd}$  deconvolution layers, where high-level information is well-preserved. Then, the probabilistic corrections  $\Delta\Psi^{t-1}$  generated from the shallow-level and deep-level layers of the network are computed and added to the decision pathway with the supervision of multiple DSLs.

As shown in Fig. 2, during inference, score maps are first initialized on the decision pathway through Eq. (3), then concatenated with the layers in the information pathway through Eq. (4). These newly formed features are processed and aggregated into the decision pathway using Eq. (5), and finally the score maps in the decision pathway are updated by Eq. (6). The same process repeats several times to generate the final score maps. The intention of this architecture is identical to cascaded regression, where in each stage, features are generated and contribute to reduce residual errors between predicted key-point locations and ground-truth locations. The predicted locations then get updated and are used to re-initialize a new stage. The difference is our 2D inference model fully exploits the information encoded in a single network instead of resorting to a stacked architecture.

**Network structure:** In Figure 2, we employed a standard DeconvNet architecture containing ten groups of layers (G1, G2, G3, G4, G5, G6, G7, G8, G9, G10) as feature source. Each group contains two or three convolutional/deconvolutional layers, batch normalization layers, and one pooling/unpooling layer.

We added a hyperlink to connect G4 and G8 to avoid information bottleneck. The decision pathway is derived from the layer of G8, before unpooling. Bilinear upsampling layers are denoted as U1 and U2. Loss layer SL represents a degraded DSL (introduced in Section 3.3, represented as SL). We use P-DSL to represent DSL used for supervising key-point candidate proposal. We use R-DSL to represent DSL used for supervising candidate refinement. Shapes of these DSLs are plotted in Fig. 1. The layers G12, G13, G14, G15, and G16 are additional groups of layers used to convert feature maps from information pathway to score maps in the decision pathway. The layers G12 and G13 contain three convolutional and two batch normalization layers. The layers G14, G15, and G16 include one convolutional layer. The settings of convolutional layers in G12 and G13 are the same: width 3, height 3, stride 1, pad 1 except the converters (last layer of G12 and G13) which connect the information pathway and the decision pathway, whose kernel size is  $1 \times 1$ . The other converters G14, G15, and G16 have the same kernel size:  $1 \times 1$ . The whole network takes 1GB GPU memory.

#### 4. Experiments

In our experiments, we train the network from scratch. For each score map, there is only one pixel at most that is marked as key-point pixel (depending on the visibility of the key-point). We employ different sampling ratios for the background pixels that are nearby or further from the key-point. The threshold for differentiating ‘nearby’ and ‘far-away’ is measured by pixel distance on the score maps. In this paper, we employ three pixels as the threshold. At the beginning, the network is trained with features generated from shallow-level layers only, which means the network has three loss functions instead of five in the first three epochs. After training the network for three epochs, we fine-tune the network with all five loss functions for another three epochs. In these six epochs, we employ a degraded DSL (SL) as explained in Section 3.2, then DSL is used and the whole architecture is as shown in Fig. 2. The learning rate is gradually reduced from  $10^{-3}$  to  $10^{-7}$  during the whole training process. We employ the stochastic gradient descent method (SGD) to train the network. The input size of the network is  $160 \times 160$  (gray scale) and the output size of score map is  $80 \times 80$ . It takes three days to train on one NVIDIA Titan X. The detailed parameter settings in training are shown in Tables 2-6.



Table 2: Parameters of SL.

Stage	Sampling ratio: Far-away pixels	Sampling ratio: Nearby pixels	Value of $\alpha$	Value of $\beta$	Type of Loss	Epoch
1	0.005	0.1	1	0.2	SL	3
2	0.005	0.1	1	0.2	SL	3
3	0.005	0.1	1	0.2	SL	3

Table 3: Parameters of P-DSL1.

Stage	Sampling ratio: Far-away pixels	Sampling ratio: Nearby pixels	Value of $\alpha$	Value of $\beta$	Type of Loss	Epoch
1	0.005	0.1	1	0.2	SL	3
2	0.001	0.2	3	0.1	SL	3
3	0.001	0.15	3	0.6	DSL	3

Table 4: Parameters of R-DSL1.

Stage	Sampling ratio: Far-away pixels	Sampling ratio: Nearby pixels	Value of $\alpha$	Value of $\beta$	Type of Loss	Epoch
1	-	-	-	-	-	-
2	0.01	0.05	1	0.3	SL	3
3	0.01	0.05	1.5	1	DSL	3

Table 5: Parameters of P-DSL2.

Stage	Sampling ratio: Far-away pixels	Sampling ratio: Nearby pixels	Value of $\alpha$	Value of $\beta$	Type of Loss	Epoch
1	0.005	0.1	1	0.2	SL	3
2	0.001	0.2	3	0.1	SL	3
3	0.001	0.15	3	0.6	DSL	3

Table 6: Parameters of R-DSL2.

Stage	Sampling ratio: Far-away pixels	Sampling ratio: Nearby pixels	Value of $\alpha$	Value of $\beta$	Type of Loss	Epoch
1	-	-	-	-	-	-
2	0.01	0.05	1	0.3	SL	3
3	0.01	0.05	1.5	1	DSL	3

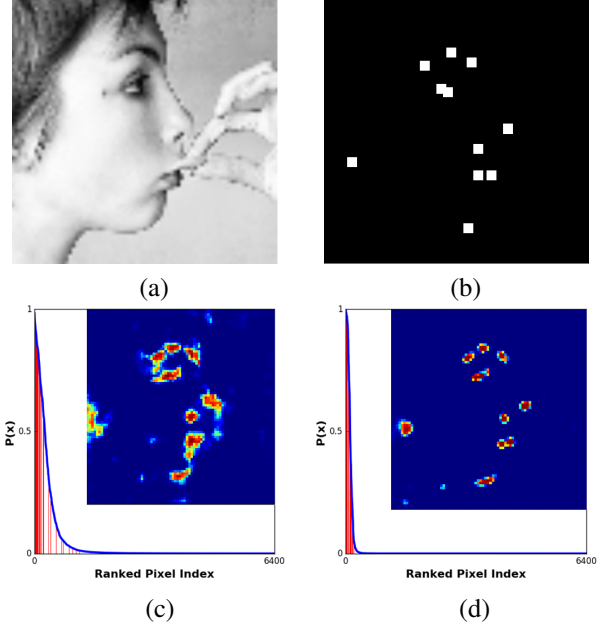


Figure 4: Score maps generated from different variations of DeconvNet. Pixels in the score maps indicate probabilities of visible facial key-points. (a) Original image, (b) Ground-truth mask, (c) DeconvNet [29], (d) GoDP. We rank the pixel values in each score map and plot as the blue curve line underneath. The red vertical lines indicate the pixel values in the key-point candidate positions ( $3 \times 3$  white patches plotted in (b)). This comparison shows that the score map generated from GoDP is clear and discriminative (best viewed in color).

#### 4.1. Databases and baselines

Three highly challenging databases are employed for evaluation: AFLW [21], AFW [22], and UHDB31 [38]. The detailed experimental settings are summarized in Table 1. We strictly follow the training and testing protocol as in [2], and conduct our experiment on AFLW-PIFA (3,901 images for training, 1,299 images for testing, 21 landmarks annotated in each image) and ALFW-Full (20,000 training, 4,386 testing, 19 landmarks annotated in each image). We note the models trained on AFLW-PIFA as  $M1$ , and the models trained on AFLW-Full as  $M2$ . For evaluating on AFW database (468 images for testing, six landmarks annotated in each image),  $M2$  is used. We picked six estimated landmarks out of 19 to report the performance. To evaluate the accuracy of algorithms under frontal faces,  $M2$  is also employed. Different from [2], all 1,314 images out of 4,386 in AFLW-Full database with 19 visible landmarks are considered as frontal faces and used for testing. Results are shown in Table 9 with the name AFLW-F. The database UHDB31 is a lab-environment database which contains 1,617 images, 77 subjects, and 12 annotated landmarks for each image. This is a challenging database including 21 head poses, combining seven yaw variations:  $[-90^\circ: +30^\circ: 90^\circ]$  and three pitch variations:  $[-30^\circ: +30^\circ: 30^\circ]$ . We employed nine landmarks (ID: 7,9,10,12,14, 15,16, 18,20 in [21]) to compute landmark errors. Model  $M2$  is employed for evaluating.

Multiple state-of-the-art methods (CDM [17], RCPR [15], CFSS [19], ERT [39], SDM [16], LBF [18], PO-CR [1], CCL

[2], CALE [40], HF [10], PAWF [4], 3DDFA [3]) are selected as baselines. In our implementation, Hyperface (HF) is trained without the loss of gender. The network architecture remains the same. The performance of 3DDFA and PAWF are reported based on the code provided by their authors. We employed normalized mean error (NME) to measure the performance of algorithms as in [2]. Same as [2], the bounding box defined by AFLW is used to normalize the mean error of landmarks and initialize the algorithm. When the AFLW bounding box is not available (*e.g.*, on UHDB31 and AFW database) or not rectangle (AFLW-PIFA), we use the bounding box generator provided by the authors of AFLW to generate a new bounding box based on the visible landmarks. For AFLW-PIFA database, after we generate new bounding boxes, we enlarge them by 15% to guarantee the ears are included, while the NME is computed using the original size of the bounding boxes.

#### 4.2. Architecture analysis and ablation study

Along with the development of the deep network, the network structures become very complex, which might make the functionality of individual modules unclear to the reader. To

Table 7: Performance on PIFA-AFLW database. MPK (%) represents mean probability of key-point candidate pixels (large is better). MPB (%) represents mean probability of background pixels (small is better). NME-Vis represents NME (%) of visible landmarks. NME-All represents NME (%) of all 21 landmarks.

Method	MPK	MPB	NME-Vis	NME-All
DeconvNet [29]	51.83	4.50	4.13	8.36
HGN [30]	28.38	0.96	3.04	11.05
GoDP-DSL-PR	31.79	1.01	3.35	13.30
GoDP-DSL	26.15	0.86	3.87	13.20
GoDP	39.78	1.30	2.94	11.17
GoDP(A)-DSL	99.37	99.92	3.37	5.75
HGN(A)	48.42	14.19	3.08	5.04
GoDP(A)	47.59	6.74	<b>2.86</b>	<b>4.61</b>

evaluate different networks’ capability for generating discriminative score maps, we analyzed new connections/structures of recent architectures on the DeconvNet platform [29] to control uncertainty. The hour-glass network (HGN) [30] is a recent

Table 8: Analysis of pose subspace partition on PIFA-AFLW and AFLW-Full database. Along with the results of GoDP(A) we present the most recent state-of-the-art methods on these datasets. The number of partitions employed in the works is denoted by  $K$ . The NME (%) of all landmarks are reported in the databases.

Method	Evaluation	K=1	K=3	K=5	K=16
CCL [2]	PIFA-AFLW	-	-	-	5.81
CALE-detector [40]	PIFA-AFLW	5.53	-	-	-
CALE-regressor [40]	PIFA-AFLW	4.38	-	-	-
GoDP(A)	PIFA-AFLW	<b>4.33</b>	4.61	-	-
CCL [2]	AFLW-Full	3.73	-	-	2.72
DAC-CSR [41]	AFLW-Full	-	-	2.08	-
GoDP(A)	AFLW-Full	1.93	<b>1.84</b>	-	-

extension of DeconvNet. The core contribution of Hour-glass net is that it aggregates features from shallow to deep layers through hyper-connections, which blends the spatial and semantic information for discriminative localization. Different from our supervised proposal and refinement architecture, the information fusion of HGN is conducted in an unsupervised manner. Our implementation of Hour-glass net is based on DeconvNet, we add three hyper-links to connect shallow and deep layers but remove residual connections [42]. This model is selected to be our baseline. The detailed network settings for our implementation can be viewed in Fig. 3.

In this experiment, we first employ a landmark mask to separate foreground and background pixels as shown in Fig. 4. Then we compute the mean probability of foreground and background pixels based on the mask. We average the mean probability over all the testing images on the PIFA-AFLW database and obtain the numbers in Table 7. We observed that GoDP performs significantly better in discriminating foreground and background pixels than other structures and has a smaller landmark detection error. We also evaluated our architecture without DSL (shows as ‘-DSL’ in Table 7), and another architecture without both DSL and proposal-refinement (PR) architecture (degraded DSL everywhere with the same parameters). The re-

Table 9: NME (%) of visible landmarks on multiple database partitions. The deep learning based methods are mainly compared in this table.

Evaluation	Deep Cascaded R.		Deep End-to-End	
	PAWF	3DDFA	HF	GoDP(A)
AFLW-PIFA	4.04	5.42	-	<b>2.86</b>
AFLW-Full	-	4.52	3.60	<b>1.64</b>
AFLW-F	-	4.13	2.98	<b>1.48</b>
AFW	4.13	3.41	3.74	<b>2.12</b>

Table 10: Performance of GoDP(A)/HF on 21 views of UHDB31. NME (%) of visible landmarks is reported. Columns correspond to pitch variations, rows correspond to yaw variations.

30°	<b>2.0/5.2</b>	<b>2.1/4.9</b>	<b>1.8/5.8</b>	<b>1.6/4.2</b>	<b>1.8/5.5</b>	<b>2.1/5.2</b>	<b>1.8/5.3</b>
0°	<b>1.8/3.6</b>	<b>1.8/3.0</b>	<b>1.8/3.2</b>	<b>1.2/2.2</b>	<b>1.5/3.0</b>	<b>1.7/3.3</b>	<b>1.7/3.7</b>
-30°	<b>2.7/5.3</b>	<b>2.1/5.3</b>	<b>1.8/4.3</b>	<b>1.6/3.3</b>	<b>1.6/3.8</b>	<b>2.1/5.0</b>	<b>2.4/6.0</b>
	-90°	-60°	-30°	0°	30°	60°	90°

sult is as shown in Table 7 with the name ‘GoDP-DSL-PR’. Table 7 shows that DSL is critical for training a highly discriminative key-point detector and also contributes to regularize our PR architecture. Additionally, we observe that the hyper-links introduced in Hour-glass net suppress background noise in DeconvNet.

In the next experiment, we trained GoDP to detect occluded landmarks. In stage 3, we used the coordinates of all landmarks as the ground-truth of the last two DSLs (previous DSL/SL are trained with visible landmarks), fine-tuned from stage 2, and trained the whole network for three epochs. The results are shown in Table 7 with the name **GoDP(A)**. To compare with HGN, we trained the HGN to detect occluded landmarks, the result is as shown as HGN(A). We observe that GoDP(A) can better detect both visible and invisible (all) landmarks if we train the network in this manner. Since then, we use GoDP(A) in the following experiments for comparison.

#### 4.3. Analysis of subspace partition

Since pose encodes multi-modality variations of facial appearance, several recent works divide the target space of landmark detection into  $K$  subspaces according to pose. We analyzed the effects of  $K$  on two databases. The results are shown in Table 8. Three recent state-of-the-art methods are employed as baseline: CCL [2], DAC-CSR [41], and CALE [40]. The target space partition strategy is employed in CCL and DAC-CSR. We tested the performance of GoDP(A) under  $K=1$  and  $K=3$  ( $K=3$  is our default configuration in all the experiments). The results show that GoDP(A) behaved differently on the two databases. Note that on PIFA-AFLW,  $K=1$  yields better performance, however, on AFLW-Full,  $K=3$  outperforms  $K=1$  and yields the best result. This phenomenon might be caused by the amount of training data. AFLW-Full includes many training samples than PIFA-AFLW and excludes two ambiguity points under both ears. These advantages help to train a stable and accurate model per subspace.



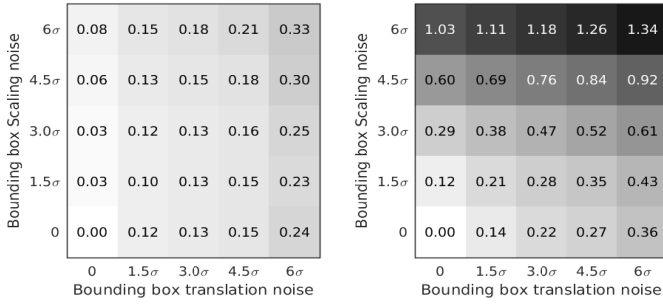


Figure 6: Robustness evaluation. NME (%) increasing on all 19 landmarks of AFLW-Full database 4,386 images under noised bounding box initializations. The  $\sigma$  is measured in percentage. (L) GoDP(A). (R) HF.

#### 4.4. Performance on challenging databases

We compare GoDP(A) with state-of-the-art cascaded regression based key-point detectors and report the performance of detecting visible and all landmarks in Tables 9-11, and Fig. 5. Since we strictly follow the experimental protocol as in [2], we directly cite their numbers in Table 11.

We first compare the performance of four deep learning based methods in detecting visible landmarks on challenging databases. In Table 9, we observe that GoDP(A) performs the best, HF ranked the second. Because HF relies on a global mapping from image ROI to shape space, it is not as discriminative as GoDP(A) in terms of localizing exact key-point positions. Deep cascaded regression methods like PAWF and 3DDFA performed well but not as accurately as GoDP(A). This observation can be further demonstrated by detecting landmarks on frontal faces. On the AFLW-F database, GoDP(A) performed significantly better than HF and 3DDFA on Table 9. In Table 10, we compare the performance of GoDP(A) and HF on 21 views of UHDB31 in terms of the NME of visible landmarks. We observe that GoDP(A) exhibits viewpoint-invariant properties. Only in extreme views does the landmark detection performance have a slight degradation. A qualitative result of this property can be viewed in Fig.13 and Fig.14.

Only comparing the NME of visible landmarks is not comprehensive enough. A good landmark detector must be capable of precisely estimating the positions of both visible and invisible landmarks. We report the NME of all annotated landmarks in Table 11. In Table 11 we observe that when training database (20,000 images) is large enough, non-deep learning based methods can also perform competitively. AFLW-Full has almost five times more training samples than AFLW-PIFA; it is obvious that the performance of RCPR, CFSS, and CCL is significantly improved when training on a larger database. Deep cascaded regression PAWF and 3DDFA did not perform as well as CCL, which may be caused by the precision weakness of deep regression and indirect mapping from 3D pose parameters to 2D landmarks.

We plot the CED curve of deep learning based methods in Fig. 5 and present more qualitative results in Fig. 12-14. Even though GoDP(A) outperforms the others, from Fig. 5(b) we ob-



Figure 7: (T) 21 views of UHDB31 database. Results are reported based on the head pose in Fig. 8 and Fig. 10. (B) The center view is selected as gallery, the other 20 views are selected as probe. Each probe image is compared with all the 77 frontal gallery images in the database to determine its identity.

serve that the performance of GoDP(A) and HF is not as consistent as 3DDFA and PAWF in terms of detecting both visible and invisible landmarks. Because 3DDFA and PAWF rely on parametric 3D shape model to localize 2D landmarks, 3D shape provides a stronger geometric regularization over the invisible landmarks. In contrast, because GoDP(A) and HF directly estimate the landmark locations. They are relatively more sensitive to self-occlusions. This trade-off between detection accuracy and shape rationality requires more exploration in future work.

To further review the properties of GoDP, we compare robustness of GoDP(A) and HF (regression-based method) under different bounding box initializations. This is important because bounding boxes generated by real face detectors always vary in size and position. We artificially add Gaussian noise to the provided bounding boxes of AFLW-Full. The noise is generated based on the size of bounding boxes, where  $\sigma$  controls the intensity of the Gaussian noise. The noise is added on the size and location of bounding boxes, and the results are as shown in Fig. 6, which discloses that GoDP(A) is much more to the initialization of bounding boxes than HF. It also discloses that GoDP(A) is robust to variations of bounding box sizes but sensitive to translation errors. One explanation is that because GoDP(A) is a detection-based method, it is unable to predict any key-points outside the response region, but regression based methods can. One solution to compensate for this limitation in future is through randomly initializing multiple bounding boxes as in ERT [39] and predicting landmark locations using median values.

#### 4.5. Performance in 3D-aided face recognition system

In the previous experiments we demonstrated the proposed landmark detector could accurately annotate facial fiducial points in challenging poses. In this subsection, we compare the performance of GoDP(A) with a state-of-the-art landmark detector

Table 11: NME (%) of all annotated landmarks.

Baseline	Non-Deep Learning methods								Deep Cascaded R.		Deep End-to-End	
	CDM	RCPR	CFSS	SDM	LBF	PO-CR	CCL	DAC-CSR	PAWF	3DDFA	HF	GoDP(A)
AFLW-PIFA	8.59	7.15	6.75	6.96	7.06	-	5.81	-	6.00	6.38	-	<b>4.61</b>
AFLW-Full	5.43	3.73	3.92	4.05	4.25	5.32	2.72	2.08	-	4.82	4.26	<b>1.84</b>

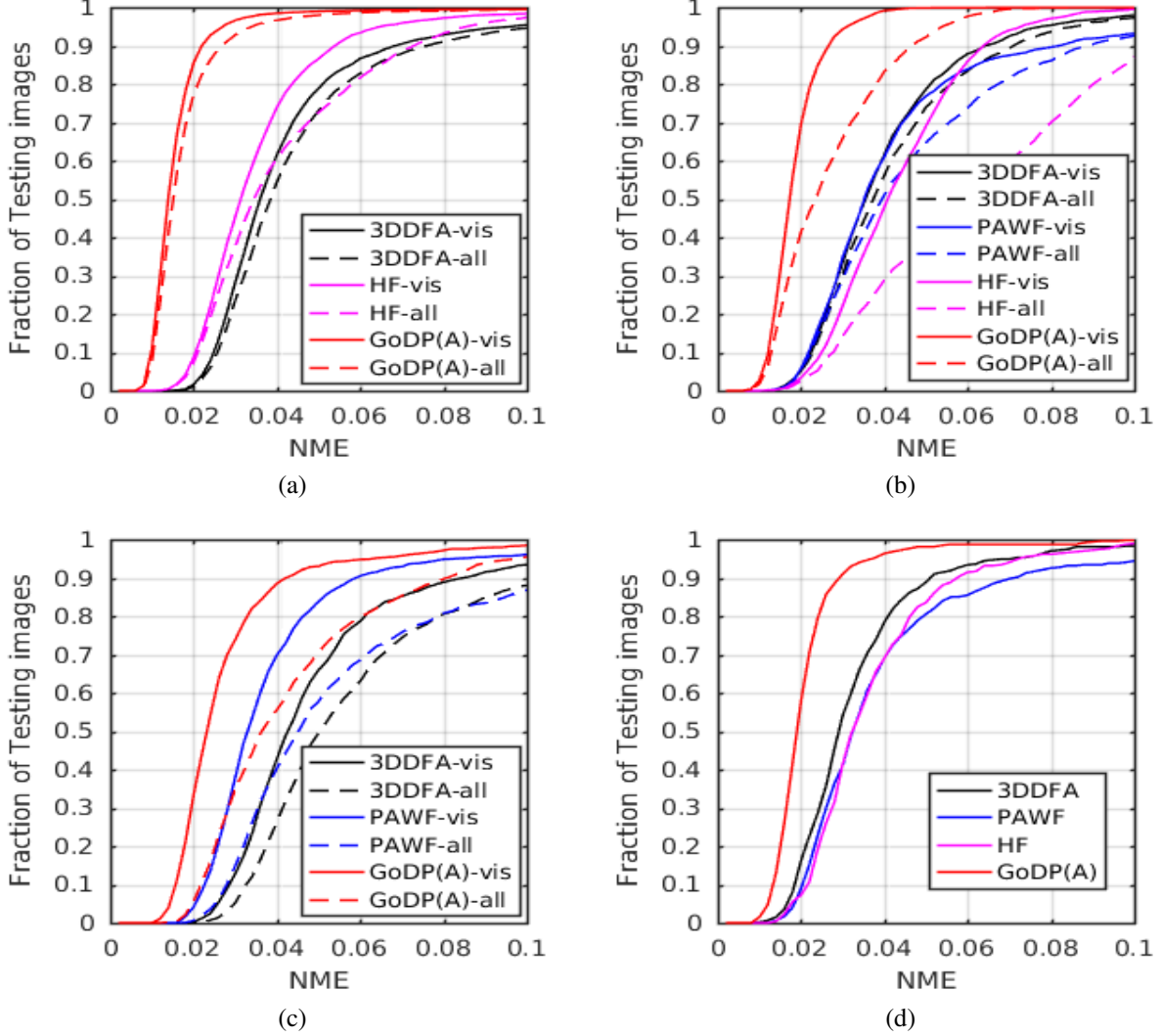


Figure 5: The CED of deep learning based methods. ‘vis’/‘all’ represents the error of visible/all landmarks. (a) AFLW-Full: 4,386 images, 19 landmarks. (b) UHDB31: 1,617 images, 9 landmarks. (c) AFLW-PIFA: 1,299 images, 21 landmarks. (d) AFLW: 468 images, 6 landmarks. GoDP outperforms the other methods in all databases.

ERT [39] (implemented in Dlib [43]) in a 3D-aided face recognition pipeline [44]. We show that our proposed method significantly improves the accuracy of facial texture frontalization accuracy in terms of rank-1 identification rate. Two deep learning based face identification approaches, Openface [12] and VGG face descriptor [13], are evaluated in the same database. We demonstrate that through accurate 3D model based face frontalization, signatures generated by traditional face descriptors could outperform well-trained deep representations.

We employ a 3D-aided face identification pipeline proposed by Kakadiaris *et al.* [44, 45]: UR2D. UR2D uses 3D models to transform facial textures into canonical UV spaces, where facial pose is normalized to frontal. The original UR2D pipeline assumes a personalized 3D model is captured for each subject in the gallery. However, this setting is not easy to satisfy in real application. Hence, we employ an extension of the pipeline

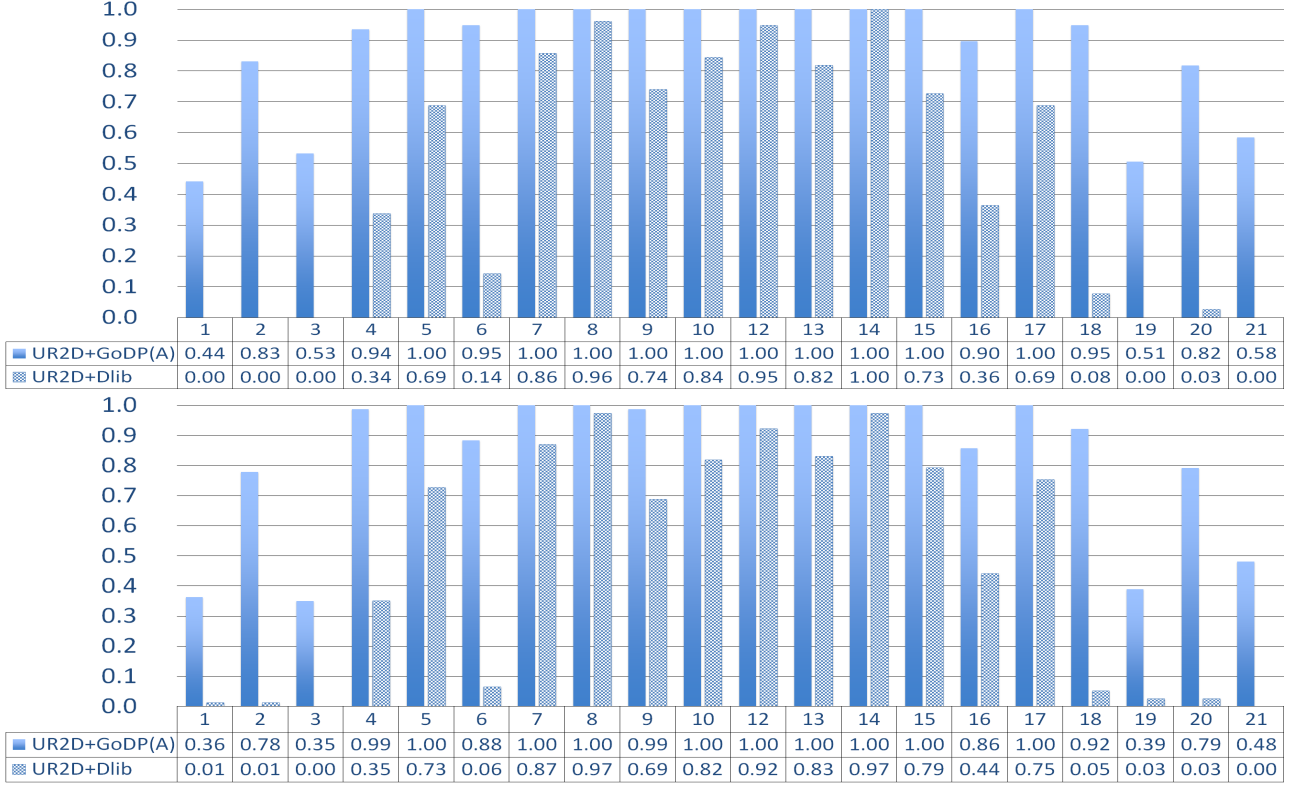


Figure 8: Rank-1 identification rate of 20 poses on (T)  $256 \times 306$  and (B)  $128 \times 153$  resolution resolution.

proposed by Dou *et al.* [46]<sup>1</sup>, which employs 2D facial landmarks to reconstruct a personalized 3D model for both gallery and probe during matching. Following the 1-vs-all experimental protocol [38] on UHDB31 database, we select 77 images (1 image per subject) in frontal view as gallery and all the other views of UHDB31 database (77 subjects  $\times$  20 poses = 1540 images in total) as probe. A sample of this evaluation protocol is shown in Fig. 7. In the experiment, we compute the similarity scores between a probe image with all 77 gallery images, and determine the ID of the probe based on the 77 similarity scores. Since the original images in UHDB31 are in high resolution ( $1024 \times 1224$ ), we down-sampled all the images into 1/4 and 1/8 sizes and report our pipeline’s performance on the two resolutions. Note that we did not use any images in UHDB31 to train UR2D.

Our pipeline is fully automatic, the input is a raw facial image and the output is a facial signature. During matching, we first detect faces based on DPM [47] head hunter, which is able to detect more than 95% faces on UHDB31 database on both resolutions. Then, a pre-trained deep bounding box regressor (similar to [48]) is employed to refine the facial ROIs to approximate the bounding boxes defined in AFLW—the database we used to train our landmark detector. We randomly initialize 16 ROIs with slight perturbations from the refined DPM bounding box, then employ GoDP(A) (trained on AFLW-Full

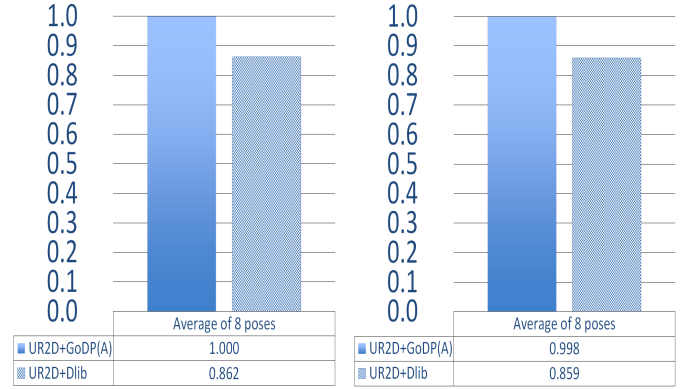


Figure 9: Average rank-1 identification rate of center eight poses on (L)  $256 \times 306$  resolution and (R)  $128 \times 153$  resolution.

database) to localize facial landmarks on each of them. The median is employed to summarize predictions on the 16 perturbations. After obtaining 19 landmarks for each facial image, we reconstruct a personalized 3D facial model based on 2FCSL [49], then use this generated 3D facial model to lift facial textures from the images [46]. In these stages, facial landmarks play a critical role since they directly influence the quality of constructed 3D models and frontalized facial textures. After obtaining the lifted textures, we extracted 65,536 dimensions DFD signature [11] on the frontalized facial texture as proposed

<sup>1</sup>We directly employ the code, trained models provide by the original authors.

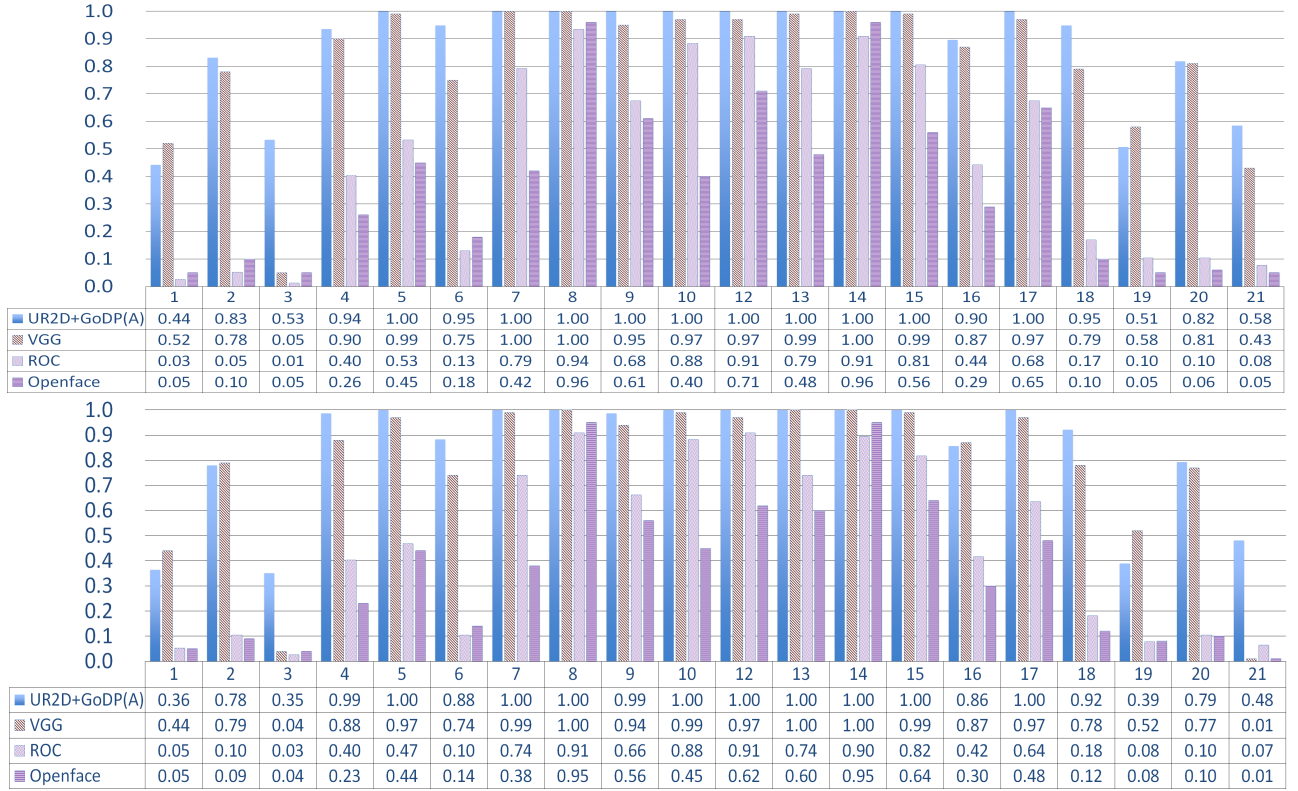


Figure 10: Rank-1 identification rate of 20 poses on (T)  $256 \times 306$  resolution and (B)  $128 \times 153$  resolution.

in [46]. Note that DFD [11] is a traditional image patch based discriminant descriptor; the authors of [46] trained it on FRGC v2 [50] database with only 907 images. The cosine distance of two DFD signatures is used to represent the similarity between each pair of gallery and probe image.

We first compare the improvement of face identification with Dlib landmark detector on UHDB31. We employed the default face detector in Dlib library to initialize the Dlib landmark detector to keep its performance. The rank-1 identification rate of the pipeline is employed to measure the accuracy of face alignment. The results are shown in Fig. 8. Since Dlib landmark detector is only trained on near-frontal poses, the results of pose 8 to 15 (except 11) are considered to be fair comparisons. In these eight views, as shown in Fig. 9, the rank-1 face identification rate of our pipeline is 99.8% on 128 resolution and 100% on 256 resolution, and has a 13.9%/13.8% improvement over the Dlib landmark detector. This experiment demonstrates that we are able to improve rank-1 identification rate further on near-frontal views with more accurate landmark detection.

We then compare our pipeline with other face identification systems in terms of rank-1 identification rate. For the VGG face descriptor, following [13], we feed it the cropped faces generated by DPM face detector directly. We did not use any 2D affine transformation to normalize the faces for VGG because it strongly distorts the faces in extreme views and degrades identification rates. For OpenFace, we used the bounding boxes generated by Dlib face detector to crop the faces at first, but when Dlib failed, we used the DPM head hunter. This guarantees the

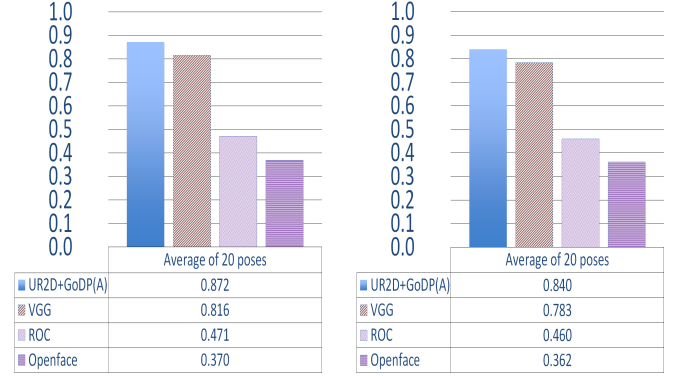


Figure 11: Average rank-1 identification rate of 20 poses on (L)  $256 \times 306$  resolution and (R)  $128 \times 153$  resolution.

number of detected faces are identical in comparison. Following the original implementation in [12], Dlib landmark detector and affine transformation are implemented before generating the facial signature. In addition to VGG and Openface, we employ a commercially available face recognition system named Rank One Computing (ROC) [51] as another baseline. This system comes along with pre-trained face detection and signature generation algorithms. We report the rank-1 identification rate in Fig.10 and Fig.11. We observe that our face recognition system significantly outperforms Openface and ROC on all the head poses, and better than VGG descriptor in most of the head



poses. In the two extreme poses, pose 3 and pose 21, our approach still obtains more than 50% identification rate at 256 resolution.

This demonstrates that, with good face alignment, a shallow descriptor trained on a few thousand images is able to perform very competitively. An explanation for VGG and DPM perform poorly on UHDB31 database but much better on LFW [52] is that more than 50% of facial images in UHDB31 have head pose larger than 30 degrees in both pitch and yaw variations, while LFW is a near-frontal database [53]. In another perspective, the experiment demonstrates that state-of-the-art deep features are still vulnerable under large head pose variations even if the database is relatively small (77 subjects). In this case, good face alignment is essential.

## 5. Conclusion

We propose an efficient deep architecture that is able to localize facial key-points precisely. The architecture transforms the traditional regression problem into a 2D cascaded key-point detection problem through a new loss function and a unique proposal-refinement technique. Therefore, we successfully tackle the optimization and precision weakness of deep cascaded regression. The intense experimental analysis shows the benefits of the proposed architecture over multiple state-of-the-art deep and shallow structures. Finally, we demonstrate that our landmark detector can significantly improve the quality of face frontalization in 3D-aided face identification.

## 6. Acknowledgment

The authors would like to thank Le Anh Vu Ha and Pengfei Dou to provide experimental data for face recognition. This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2015-ST-061-BSH001. This grant is awarded to the Borders, Trade, and Immigration (BTI) Institute: A DHS Center of Excellence led by the University of Houston, and includes support for the project “Image and Video Person Identification in an Operational Environment: Phase I” awarded to the University of Houston. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

## References

- [1] G. Tzimiropoulos, Project-out cascaded regression with an application to face alignment, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts, 2015, pp. 3659–3667.
- [2] S. Zhu, C. Li, C. C. Loy, X. Tang, Unconstrained face alignment via cascaded compositional learning, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 3409–3417.
- [3] X. Zhu, Z. Lei, X. Liu, H. Shi, S. Z. Li, Face alignment across large poses: A 3D solution, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 146–155.
- [4] A. Jourabloo, X. Liu, Large-pose face alignment via CNN-based dense 3D model fitting, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 4188–4196.
- [5] H. Qin, J. Yan, X. Li, X. Hu, Joint training of cascaded CNN for face detection, in: Proc. Computer Vision and Pattern Recognition, Las Vegas, 2016.
- [6] X. Peng, R. S. Feris, X. Wang, D. N. Metaxas, A recurrent encoder-decoder for sequential face alignment, in: Proc. European Conference on Computer Vision, Amsterdam, Netherlands, 2016, pp. 38–56.
- [7] G. Ghiasi, C. C. Fowlkes, Laplacian pyramid reconstruction and refinement for semantic segmentation, in: Proc. European Conference on Computer Vision, Amsterdam, Netherlands, 2016, pp. 519–534.
- [8] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: Proc. Computer Vision and Pattern Recognition, Portland, Oregon, 2013, pp. 3476–3483.
- [9] Z. Zhang, P. Luo, C. C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, in: Proc. European Conference on Computer Vision, Zurich, Switzerland, 2014, pp. 2715–2718.
- [10] R. Ranjan, V. M. Patel, R. Chellappa, Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition, arXiv:1603.01249.
- [11] Z. Lei, M. Pietikainen, S. Li, Learning discriminant face descriptor, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2) (2014) 289–302.
- [12] B. Amos, B. Ludwiczuk, S. Mahadev, Openface: A general-purpose face recognition library with mobile applications, Tech. Rep. CMU-CS-16-118, CMU School of Computer Science, Pittsburgh, PA (2016).
- [13] O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: Proc. British Machine Vision Conference, Vol. 1, 2015, pp. 1–12.
- [14] X. Cao, Y. Wei, F. Wen, J. Sun, Face alignment by explicit shape regression, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 2887–2894.
- [15] X. P. Burgos-Artizzu, P. Perona, P. Dollár, Robust face landmark estimation under occlusion, in: Proc. IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 1513–1520.
- [16] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Portland, Oregon, 2013, pp. 532–539.
- [17] X. Yu, J. Huang, S. Zhang, W. Yan, D. N. Metaxas, Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model, in: Proc. IEEE International Conference on Computer Vision, Sydney, Australia, 2013, pp. 1944–1951.
- [18] S. Ren, X. Cao, Y. Wei, J. Sun, Face alignment at 3000 FPS via regressing local binary features, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1685–1692.
- [19] S. Zhu, C. Li, C. C. Loy, X. Tang, Face alignment by coarse to fine shape searching, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, 2015, pp. 4998–5006.
- [20] A. Jourabloo, X. Liu, Pose-invariant 3D face alignment, in: Proc. International Conference on Computer Vision, Santiago, Chile, 2015.
- [21] M. Kostinger, P. Wohlhart, P. M. Roth, H. Bischof, Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization, in: Proc. IEEE International Workshop on Benchmarking Facial Image Analysis Technologies, Barcelona, Spain, 2011, pp. 2144–2151.
- [22] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 2879–2886.
- [23] H. Fan, E. Zhou, Approaching human level facial landmark localization by deep learning, Image and Vision Computing 47 (2016) 27–35.
- [24] J. Zhang, M. Kan, S. Shan, X. Chen, Occlusion-free face alignment: deep regression networks coupled with de-corrupt autoencoders, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 3428–3437.
- [25] X. Xu, I. A. Kakadiaris, Joint head pose estimation and face alignment framework using global and local cnn features, in: Proc. 12<sup>th</sup> IEEE Conference on Automatic Face and Gesture Recognition, Washington, DC, 2017 (In press).
- [26] A. Bulat, G. Tzimiropoulos, Convolutional aggregation of local evidence for large pose face alignment, in: Proc. British Machine Vision Conference, York, United Kingdom, 2016.
- [27] S. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 4724–4732.



Figure 12: Qualitative results on AFLW-Full database. (T) HF [10], (M) 3DDFA [3], (B) GoDP(A) with score maps.

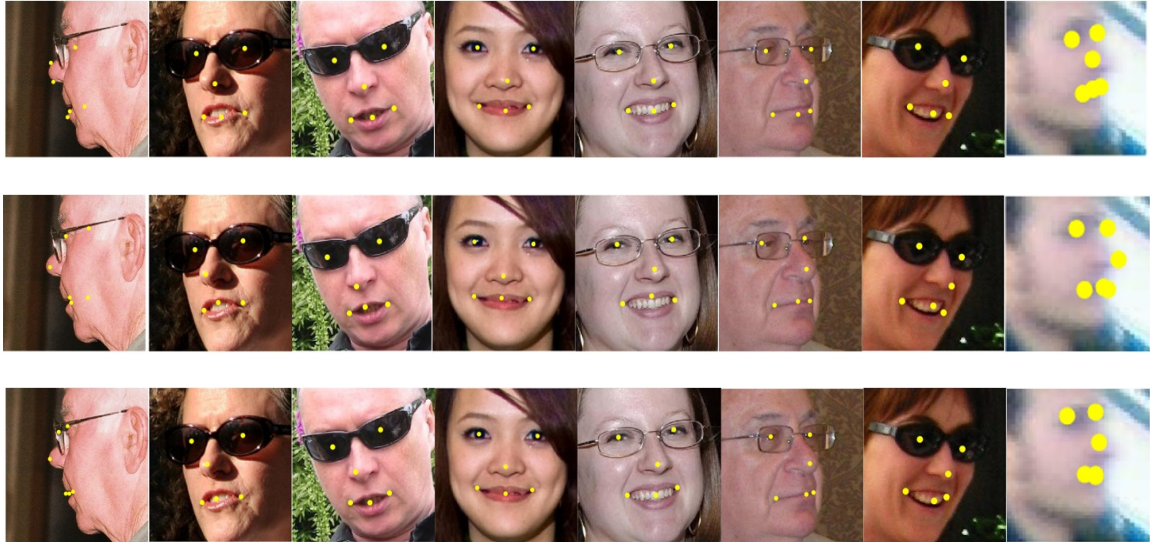


Figure 13: Qualitative results on AFW database. GoDP(A) is able to localize key-points precisely. (T) HF [10], (M) 3DDFA [3], (B) GoDP(A).

- [28] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, A. Kassim, Robust facial landmark detection via recurrent attentive-refinement networks, in: Proc. European Conference on Computer Vision, Amsterdam, Netherlands, 2016, pp. 57–72.
- [29] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proc. IEEE International Conference on Computer Vision, Santiago, Chile, 2015, pp. 1520–1528.
- [30] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: Proc. European Conference on Computer Vision, Amsterdam, Netherlands, 2016, pp. 483–499.
- [31] P. O. Pinheiro, T. Lin, R. Collobert, P. Dollar, Learning to refine object segments, in: Proc. European Conference on Computer Vision, Amsterdam, Netherlands, 2016, pp. 75–91.
- [32] X. Chu, W. Ouyang, H. Li, X. Wang, CRF-CNN: Modelling structured information in human pose estimation, in: Proc. Neural Information Processing Systems, Barcelona, Spain, 2016.
- [33] W. Yang, W. Ouyang, H. Li, X. Wang, End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 3073–3082.
- [34] J. Carreria, P. Agrawal, K. Fragkiadaki, J. Malik, Human pose estimation with iterative error feedback, in: Proc. Computer Vision and Pattern Recognition, Las Vegas, 2016, pp. 4733–4742.
- [35] Y. Wu, X. Xu, S. K. Shah, I. A. Kakadiaris, Towards fitting a 3D dense facial model to a 2D image: A landmark-free approach, in: Proc. International Conference on Biometrics: Theory, Applications and Systems, Arlington, VA, 2015, pp. 1–8.
- [36] X. Chu, W. Ouyang, H. Li, X. Wang, Structured feature learning for pose estimation, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 4715–4723.
- [37] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proc. International Conference on Machine Learning, Montreal, Quebec,





Figure 14: Qualitative results on UHDB31 database. (T) HF [10], (M) 3DDFA [3], (B) GoDP(A).

- 2009, pp. 41–48.
- [38] Y. Wu, S. K. Shah, I. A. Kakadiaris, Rendering or normalization? An analysis of the 3D-aided pose-invariant face recognition, in: Proc. IEEE International Conference on Identity, Security and Behavior Analysis, Sendai, Japan, 2016, pp. 1–8.
  - [39] V. Kazemi, J. Sullivan, One millisecond face alignment with an ensemble of regression trees, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1867–1874.
  - [40] A. Bulat, G. Tzimiropoulos, Convolutional aggregation of local evidence for large pose face alignment, in: Proc. British Machine Vision Conference, York, UK, 2016, pp. 700–712.
  - [41] Dynamic attention-controlled cascaded shape regression exploiting training data augmentation and fuzzy-set sample weighting, arXiv:1611.05396.
  - [42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 770–778.
  - [43] D. E. King, Dlib-ml: A machine learning toolkit, Journal of Machine Learning Research 10 (2009) 1755–1758.
  - [44] I. A. Kakadiaris, G. Toderici, G. Evangelopoulos, G. Passalis, X. Zhao, S. K. Shah, T. Theoharis, 3D-2D face recognition with pose and illumination normalization, Computer Vision and Image Understanding 154 (2017) 137–151.
  - [45] G. Toderici, G. Passalis, S. Zafeiriou, G. Tzimiropoulos, M. Petrou, T. Theoharis, I. A. Kakadiaris, Bidirectional relighting for 3D-aided 2D face recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 2721–2728.
  - [46] P. Dou, L. Zhang, Y. Wu, S. Shah, I. A. Kakadiaris, Pose-robust face signature for multi-view face recognition, in: Proc. Proceedings of International Conference on Biometrics: Theory, Applications and Systems, Arlington, VA, 2015.
  - [47] M. Mathias, R. Benenson, M. Pedersoli, L. V. Gool, Face detection without bells and whistles, in: Proc. European Conference on Computer Vision, Amsterdam, Netherlands, 2014, pp. 720–735.
  - [48] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Proc. Annual Conference on Neural Information Processing Systems, Amsterdam, Netherlands, 2015.
  - [49] P. Dou, Y. Wu, S. Shah, I. A. Kakadiaris, Robust 3D face shape reconstruction from single images via two-fold coupled structure learning, in: Proc. British Machine Vision Conference, Nottingham, United Kingdom, 2014, pp. 1–13.
  - [50] P. Phillips, W. Scruggs, A. O’Toole, P. Flynn, K. Bowyer, C. Schott, M. Sharpe, FRVT 2006 and ICE 2006 large-scale experimental results, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (5) (2010) 831–846.
  - [51] Rank One Computing, <http://www.rankone.io/> [online] (2017).
  - [52] G. B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the Wild: A database for studying face recognition in unconstrained environments, in: Proc. European Conference on Computer Vision Workshop on Faces in ‘Real-Life’ Images: Detection, Alignment, and Recognition, Marseille, France, 2008.
  - [53] I. Masi, S. Rawls, G. Modioni, P. Natarajan, Pose-aware face recognition in the wild, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, 2016, pp. 4838–4846.